



全国高等职业教育规划教材

# 单片机技术及应用

## (基于Proteus的汇编和C语言版)

主编 黄锡泉 何用辉

- 按照项目导向、任务驱动模式组织教材内容，融“教、学、做”于一体
- 采用C语言与汇编语言双语讲解，兼顾单片机原理，重点面向应用开发
- 引入Proteus仿真软件，教学内容上实现软硬结合、虚拟仿真学习与训练
- 配备教学课件、视频录像、仿真源码等课程资源，便于教师教学和学生自学



电子教案下载网址 [www.cmpedu.com](http://www.cmpedu.com)



机械工业出版社  
CHINA MACHINE PRESS

全国高等职业教育规划教材

# 单片机技术及应用

## (基于 Proteus 的汇编和 C 语言版)

主 编 黄锡泉 何用辉  
副主编 王麟珠 翁 伟  
          骆旭坤 林 福  
参 编 王红超 杨成菊 曾思通  
          吴云轩 罗炳莲 陈茂林  
          朱 群 刘思默 方凤玲  
主 审 林 丰 谢广友

机械工业出版社

本书按照项目导向、任务驱动的编写模式,将进行单片机应用设计与开发所必需的理论知识与实践技能分解到不同的项目和任务中由浅入深、循序渐进地讲述。本书具有3大特色:C语言与汇编语言并存,汇编语言注重硬件资源讲解,C语言注重程序开发,两者之间既可相互独立又可进行分析比较;软硬件结合、虚拟仿真,书中所有项目均以硬件实物装置展开讲解,再基于Proteus进行虚拟仿真学习训练;淡化原理、注重实用,以具体应用项目任务实现为主导,突出单片机实用技术的学习与训练。本书结构紧凑、图文并茂,配备教学课件、视频教程、仿真源码等完善的立体化课程资源光盘,具有较强的可读性、实用性和先进性。

本书既可作为高职高专院校自动化类、电子信息类、机电类和计算机类等专业的课程教材,也可作为应用型本科院校、函授学院以及相关培训班的教材,还可作为单片机应用开发人员的参考书。

## 图书在版编目(CIP)数据

单片机技术及应用(基于Proteus的汇编和C语言版)/黄锡泉,何用辉主编. —北京:机械工业出版社,2014.1(2015.7重印)

全国高等职业教育规划教材

ISBN 978-7-111-44676-7

I. ①单… II. ①黄… ②何… III. ①单片微型计算机—系统仿真—应用软件—高等教育—教材②汇编语言—程序设计—高等教育—教材③C语言—程序设计—高等教育—教材 IV. ①TP368.1②TP373③TP312

中国版本图书馆CIP数据核字(2013)第262592号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

责任编辑:王颖

责任印刷:李洋

三河市国英印务有限公司印刷

2015年7月第1版·第2次印刷

184mm×260mm·22.25印张·551千字

3001—5000册

标准书号:ISBN 978-7-111-44676-7

ISBN 978-7-89405-172-1(光盘)

定价:52.00元(含1DVD)

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

# 前 言

本书是在编者从事十多年单片机应用开发和教学改革的经验基础之上,结合单片机最新应用技术和高职高专教育最新理念,按照项目导向、任务驱动的编写模式,通过海峡两岸院校合作,共同开发编写的融合汇编语言、C语言和 Proteus 仿真教学于一体的项目式特色改革教材。本书结构紧凑、图文并茂,配备教学课件、视频教程、仿真源码等完善的立体化课程资源光盘,具有较强的可读性、实用性和先进性。

本书具有以下几个突出的特点:

1) 本书作为按照项目导向、任务驱动模式编写的工学结合特色改革教材,以典型的单片机应用项目为载体,将进行单片机应用设计与开发所必需的理论知识与实践技能分解到不同的项目和任务中由浅入深、循序渐进地讲述,体现学中做、做中学的理念,注重学生职业能力的培养。

2) 本书采用 C 语言与汇编语言双语讲解。由于汇编语言适合初学者对单片机原理与硬件资源的描述学习,语言灵活,但编程难掌握;而 C 语言编程容易掌握,适合程序开发,但适合对单片机原理与硬件方面具有一定基础者,一般面向产品开发。两者并存讲解既可相互独立学习又可进行分析比较,重点强化学生对单片机软硬件知识与编程能力的培养。

3) 本书内容软硬件结合、虚拟仿真,书中所有项目、任务均以硬件实物装置展开讲解,沿用传统单片机学习与开发的经验,又结合目前流行的单片机软硬件仿真软件 Proteus 进行项目实物装置的虚拟仿真学习与训练,适合初学者节约学习成本、提高学习兴趣和效率。

4) 本书内容选取上淡化原理、注重实用,以具体应用项目任务实现为主导,注重单片机实用技术学习与训练,有利于培养学生分析和解决实际应用项目问题的能力,强化学生项目组织与实施能力的培养,重点突出学生实践动手能力的提升。

5) 本书针对每个项目的培养目标,精心选择训练任务,体现精训精练;每个任务均可直接工程化移植使用,体现技术完整性与实用性。注重学习训练的延展性,每个任务既相对独立,又与前后任务之间保持密切的联系,由点到线,由线到面,体现知识学习与能力训练的综合性 and 系统性。

本书为福建省教育厅高等职业教育教材建设计划支持的闽台合作、工学结合的特色改革教材,以福建省先进制造业软件公共服务平台为支撑,由海峡两岸院校合作开发编写。建国科技大学黄锡泉和福建信息职业技术学院何用辉共同担任主编,负责全书内容的组织、统稿,参加编写的人员还有福建船政交通职业学院王麟珠、曾思通,黎明职业大学骆旭坤、吴云轩,闽西职业技术学院林福、罗炳莲,厦门海洋职业技术学院王红超,闽北职业技术学院杨成菊,建国科技大学陈茂林,福建信息职业技术学院翁伟、朱群、刘思默和方凤玲。本书由福建信息职业技术学院林丰教授级高工和中兴大学谢广文副教授共同主审,并对本书提出宝贵意见。在本书的编写过程中,编者参考了有关书籍及论文,并引用了其中的一些资料,在此一并向这些作者表示感谢。

本书中有些电路图为了保持与软件的统一性,使用了软件中的电路符号标准及文字描述标准,电路符号与图标不符,特此说明

限于编者的经验、水平,书中难免有不足与缺漏之处,恳请专家、读者批评指正。

编 者

# 目 录

## 前言

项目 1 单片机认知及其开发软件使用	1	3.1.4 C 语言程序分析与设计	76
任务 1.1 认知单片机及其 编程语言	1	3.1.5 基于 Proteus 的调试与仿真	88
1.1.1 初识单片机	1	任务 3.2 LED 跑马灯控制	93
1.1.2 分析单片机硬件系统	2	3.2.1 控制要求与功能展示	93
1.1.3 认知单片机编程语言	10	3.2.2 硬件系统与控制流程分析	94
任务 1.2 初步使用单片机 开发软件	12	3.2.3 汇编语言程序分析与设计	96
1.2.1 Keil 软件认知及使用	12	3.2.4 C 语言程序分析与设计	98
1.2.2 Proteus 软件认知及使用	16	3.2.5 基于 Proteus 的调试与仿真	103
随堂一练	23	随堂一练	107
项目 2 2 个 LED 发光二极管控制	25	技能训练 1: 双边拉幕灯控制	109
任务 2.1 LED 轮流闪烁控制	25	技能训练 2: 双向跑马灯控制	110
2.1.1 控制要求与功能展示	25	项目 4 LED 点阵显示控制	111
2.1.2 硬件系统与控制流程分析	26	任务 4.1 LED 按键指示灯控制	111
2.1.3 汇编语言程序分析与设计	28	4.1.1 控制要求与功能展示	111
2.1.4 C 语言程序分析与设计	38	4.1.2 硬件系统与控制流程分析	112
2.1.5 基于 Proteus 的调试与仿真	45	4.1.3 汇编语言程序分析与设计	115
任务 2.2 LED 闪烁方式控制	50	4.1.4 C 语言程序分析与设计	119
2.2.1 控制要求与功能展示	50	4.1.5 基于 Proteus 的调试与仿真	123
2.2.2 硬件系统与控制流程分析	51	任务 4.2 LED 点阵数显控制	127
2.2.3 汇编语言程序分析与设计	51	4.2.1 控制要求与功能展示	127
2.2.4 C 语言程序分析与设计	54	4.2.2 硬件系统与控制流程分析	128
2.2.5 基于 Proteus 的调试与仿真	63	4.2.3 汇编语言程序分析与设计	130
随堂一练	67	4.2.4 C 语言程序分析与设计	133
技能训练 1: 2 个 LED 闪烁控制	69	4.2.5 基于 Proteus 的调试与仿真	137
技能训练 2: 3 个 LED 闪烁控制	70	任务 4.3 键控 LED 点阵 显示控制	140
项目 3 8 个 LED 发光二极管控制	72	4.3.1 控制要求与功能展示	140
任务 3.1 LED 拉幕灯控制	72	4.3.2 硬件系统与控制流程分析	142
3.1.1 控制要求与功能展示	72	4.3.3 汇编语言程序分析与设计	142
3.1.2 硬件系统与控制流程分析	73	4.3.4 C 语言程序分析与设计	145
3.1.3 汇编语言程序分析与设计	73	4.3.5 基于 Proteus 的调试与仿真	149
		随堂一练	154

技能训练 1: 3*3 按键指示灯控制	155	随堂一练	240
技能训练 2: LED 点阵屏显示		技能训练 1: 简易方波输出控制	241
字符控制	157	技能训练 2: 测试外部脉冲频率	242
技能训练 3: 按键值显示控制	158	<b>项目 7 串行接口控制及应用</b>	244
<b>项目 5 中断系统控制及应用</b>	160	任务 7.1 串行接口分析与控制	244
任务 5.1 中断系统分析与控制	160	7.1.1 串行通信结构与功能分析	244
5.1.1 中断系统结构与功能分析	160	7.1.2 串行通信编程与控制	251
5.1.2 外部中断编程与控制	166	任务 7.2 串行转并行数显控制	252
任务 5.2 简易水情报警器控制	168	7.2.1 控制要求与功能展示	252
5.2.1 控制要求与功能展示	168	7.2.2 硬件系统与控制流程分析	253
5.2.2 硬件系统与控制流程分析	169	7.2.3 汇编语言程序分析与设计	255
5.2.3 汇编语言程序分析与设计	172	7.2.4 C 语言程序分析与设计	256
5.2.4 C 语言程序分析与设计	176	7.2.5 基于 Proteus 的调试与仿真	257
5.2.5 基于 Proteus 的调试与仿真	179	任务 7.3 单片机与 PC 串行通信	261
任务 5.3 简易地震报警器控制	185	7.3.1 控制要求与功能展示	261
5.3.1 控制要求与功能展示	185	7.3.2 硬件系统与控制流程分析	262
5.3.2 硬件系统与控制流程分析	187	7.3.3 汇编语言程序分析与设计	263
5.3.3 汇编语言程序分析与设计	190	7.3.4 C 语言程序分析与设计	265
5.3.4 C 语言程序分析与设计	193	7.3.5 基于 Proteus 的调试与仿真	266
5.3.5 基于 Proteus 的调试与仿真	196	随堂一练	272
随堂一练	201	技能训练 1: 串口控制跑马灯	273
技能训练 1: 中断加减计数器	203	技能训练 2: 双机通信控制	274
技能训练 2: 中断嵌套数显控制	204	<b>项目 8 并行 I/O 口扩展控制</b>	276
<b>项目 6 定时/计数器控制及应用</b>	206	任务 8.1 单片机并行扩展分析	276
任务 6.1 定时/计数器分析与控制	206	8.1.1 51 单片机三总线分析	276
6.1.1 定时/计数器结构与功能分析	206	8.1.2 并行 I/O 接口扩展认知	278
6.1.2 定时/计数器编程与控制	212	任务 8.2 简单并行 I/O 口	
任务 6.2 简易定时闹钟控制	214	扩展控制	279
6.2.1 控制要求与功能展示	214	8.2.1 控制要求与功能展示	279
6.2.2 硬件系统与控制流程分析	215	8.2.2 硬件系统与控制流程分析	279
6.2.3 汇编语言程序分析与设计	215	8.2.3 汇编语言程序分析与设计	282
6.2.4 C 语言程序分析与设计	220	8.2.4 C 语言程序分析与设计	284
6.2.5 基于 Proteus 的调试与仿真	224	8.2.5 基于 Proteus 的调试与仿真	285
任务 6.3 简易按键计数器控制	229	随堂一练	289
6.3.1 控制要求与功能展示	229	技能训练: 简单 I/O 口扩展控制	289
6.3.2 硬件系统与控制流程分析	229	<b>项目 9 A-D 转换控制及应用</b>	291
6.3.3 汇编语言程序分析与设计	231	任务 9.1 A-D 转换认知与分析	291
6.3.4 C 语言程序分析与设计	233	9.1.1 A-D 转换的初步认知	291
6.3.5 基于 Proteus 的调试与仿真	236		



9.1.2	ADC0809 及其 接口电路分析 .....	292	10.2.2	硬件系统与控制流程分析 ...	316
任务 9.2	单通道电压采集控制 ...	295	10.2.3	汇编语言程序分析与设计 ...	317
9.2.1	控制要求与功能展示 .....	295	10.2.4	C 语言程序分析与设计 ...	321
9.2.2	硬件系统与控制流程分析 ...	295	10.2.5	基于 Proteus 的调试与仿真 ...	325
9.2.3	汇编语言程序分析与设计 ...	296	随堂一练 .....	331	
9.2.4	C 语言程序分析与设计 .....	300	技能训练: 波形发生器控制 .....	332	
9.2.5	基于 Proteus 的调试与仿真 ...	303	附录 .....	333	
随堂一练 .....	308		附录 A	MCS-51 系列单片机 汇编指令表 .....	333
技能训练 1: 可调 PWM 输出控制 .....	309		附录 B	C51 关键字和常用 标准库函数 .....	337
技能训练 2: 单通道电压采集 显示控制 .....	310		一、关键字 .....	337	
项目 10 D-A 转换控制及应用 .....	312		二、C51 重要库函数 .....	338	
任务 10.1	D-A 转换认知与分析 ...	312	附录 C	Proteus 常用元器件 符号表 .....	341
10.1.1	D-A 转换的初步认知 .....	312	附录 D	程序下载器制作 及其下载 .....	345
10.1.2	DAC0832 及其接口 电路分析 .....	313	附录 E	ASCII (美国标准信息 交换码) 表 .....	348
任务 10.2	简易波形发生器控制 ...	316	参考文献 .....	349	
10.2.1	控制要求与功能展示 .....	316			

# 项目 1 单片机认知及其开发软件使用

## 知识与能力目标

- 1) 熟悉单片机的基本结构组成、引脚的功能。
- 2) 掌握内部存储器的结构、用途、地址分配和使用特点。
- 3) 理解单片机时序的相关概念。
- 4) 掌握单片机最小应用系统构成。
- 5) 初步学会 Keil 与 Proteus 软件的使用。



## 任务 1.1 认知单片机及其编程语言

### 1.1.1 初识单片机

图 1-1 为一个简单的单片机应用实物装置，在单片机的控制作用下，该实物装置可实现如下两种功能。

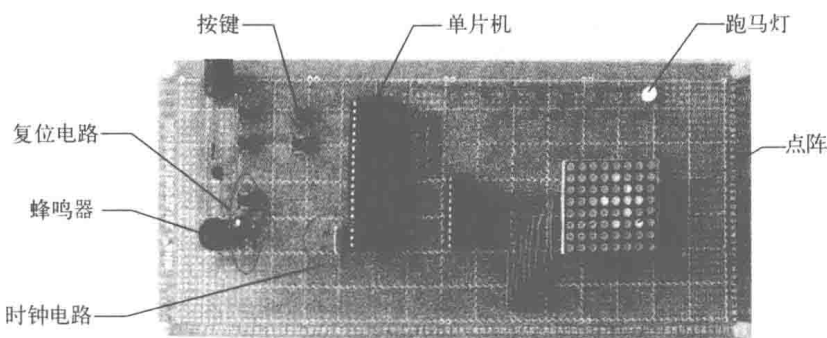


图 1-1 单片机应用实物装置

- 1) 每当有按钮按下时，蜂鸣器会发出按钮按下提示声音。
- 2) 通过 4 个按钮来控制跑马灯与点阵箭头的移动方向（左移或右移）和移动速度（加速或减速）。

该单片机应用实物装置具体的工作运行情况见教材附带光盘中的视频文件。通过图 1-1 和工作运行视频，我们发现所谓的单片机实质上就是一个芯片，通过小小一片芯片的作用即可实现复杂的控制功能。

单片机又称为单片微控制器（Single Chip Microcomputer），是采用超大规模集成电路技术把具有数据处理能力的中央处理器 CPU、随机内存 RAM、只读存储器 ROM、多种 I/O 口、



中断系统和定时/计数器等功能部件（可能还包括显示驱动电路、脉宽调制电路、模拟多路转换器和 A-D 转换器等电路）集成到一块电路芯片上构成的一个小而完善的计算机系统。

图 1-1 展示的实物装置即为单片机作为微型计算机系统的简单功能的应用，由于它具有结构简单、控制功能强、可靠性高、体积小和价格低等优点，单片机技术作为计算机技术的一个重要分支，广泛地应用于工业控制、智能化仪器仪表、家用电器和电子玩具等各个领域。

### 1.1.2 分析单片机硬件系统

自 1980 年 Intel 公司推出了 MCS-51 系列单片机，很快成为了单片机家族中的典型代表，其典型产品有 8031（内部没有程序内存）、8051（芯片采用 HMOS，功耗 630mW 是 89C51 的 5 倍）和 8751 等通用产品，实际使用中，8031 和 8051 早已经被市场淘汰。目前，以 MCS-51 技术核心为主导的单片机成为世界上许多厂家和电气公司竞选的对象，以此为基核，推出很多与 MCS-51 有极好兼容性的 CHMOS 单片机，同时还增加了许多新的功能。例如，宏晶科技公司推出的 STC89CXX 系列单片机，Atmel 公司推出的 AT89CXX 系列单片机，Philips 公司推出的 P89CXX 系列单片机，Silicon 公司推出的 C8051FXXX 系列单片机等。

#### 1. MCS-51 单片机的内部结构

MCS-51 单片机是把 CPU、RAM、ROM、定时/计数器和多种功能的 I/O 接口等功能集成在一块芯片上所构成的微型计算机，MCS-51 单片机的内部结构框图如图 1-2 所示。

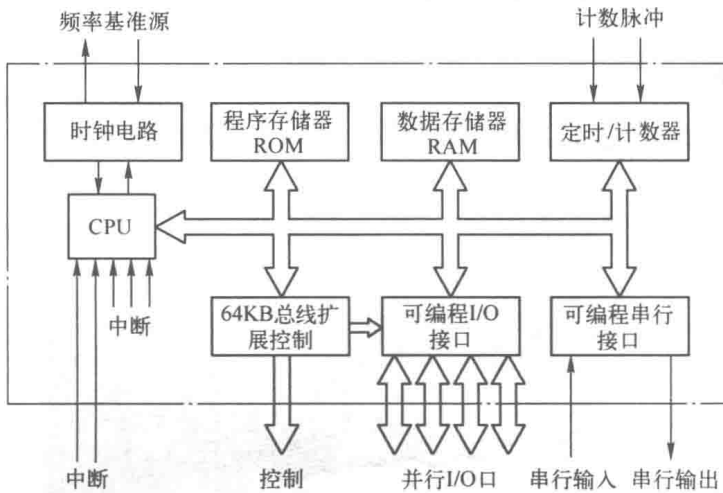


图 1-2 MCS-51 单片机的内部结构框图

- 1) CPU: CPU 是中央处理器的简称，它是单片机的核心部件，由运算器和控制器等部件组成，能够完成各种运算和控制操作。
- 2) 内存: MCS-51 单片机包括空间大小为 4KB 的片内程序内存 ROM（可扩展到 64KB）、空间大小为 256B 的片内数据存储器 RAM。
- 3) 并行 I/O 接口: MCS-51 单片机中共有 4 个 8 位并行 I/O 接口（P0、P1、P2 和 P3），每一个 I/O 接口都可以独立的用于输入和输出控制。
- 4) 定时/计数器: 51 单片机中包括两个 16 位定时/计数器。它们既可以作为定时器，用于定时、延时控制；又可以作为计数器，用于对外部事件进行计数和检测等。
- 5) 中断控制: 51 单片机具有完善的中断控制系统，其中共有 5 个中断源和两个中断优

优先级，用于满足实时控制的需要。

6) 串行接口：51 单片机采用通用异步工作方式的全双工串行通信接口，可以同时发送和接收数据。

## 2. 单片机的引脚及其功能

标准的 MCS-51 单片机是 40 引脚的芯片，本书采用的是 40 引脚双列直插 (DIP) 封装的 STC89C51 单片机，如图 1-3 所示。

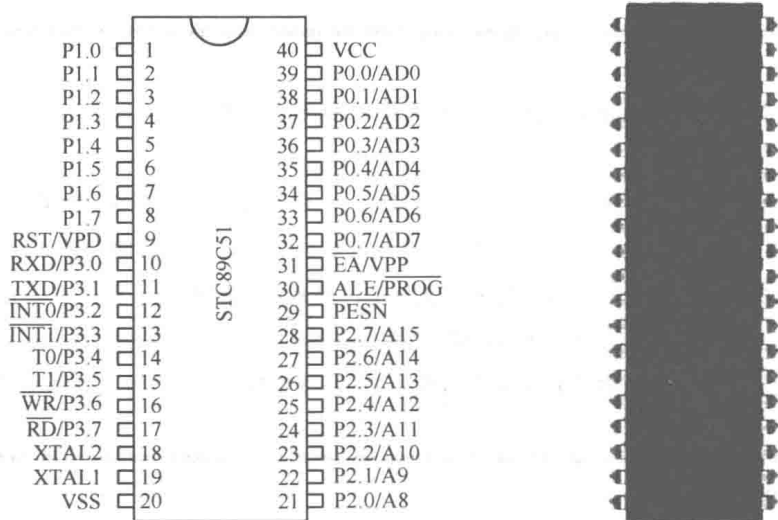


图 1-3 STC89C51 的引脚和实物图

STC89C51 单片机的 40 个引脚大致可分为电源、时钟、I/O 口和控制总线几部分，各引脚功能如下。

### (1) 电源引脚 (VCC 和 VSS)

VCC: 电源输入端，作为工作电源和编程校验。

VSS: 接共地端。

### (2) 时钟电路引脚 (XTAL1 和 XTAL2)

单片机在使用内部振荡电路时，XTAL1 和 XTAL2 用来外接石英晶体和微调电容，振荡频率为晶振频率，振荡信号送至内部时钟电路产生时钟脉冲信号。在使用外部时钟时，用于接外部时钟源。

### (3) 控制信号引脚 (RST/VPD, ALE/PROG, PSEN, EA/VPP)

1) RST/VPD: RST 复位信号输入端。当 RST 端保持两个机器周期以上的高电平时，单片机完成复位操作，VPD 为内部 RAM 的备用输入电源。当电源 VCC 一旦断电或者电压降到一定值时，可以通过 VPD 为单片机内部 RAM 提供电源，以保证片内 RAM 中信息不丢失，且上电后能够继续正常运行。

2) ALE/PROG: 地址锁存信号。在访问片外内存时，ALE 用于控制 P0 口输出低 8 位地址送入锁存器锁存起来，以实现低位地址和数据的分时传送。即使在不访问外部数据存储时，ALE 以 1/6 晶振频率的固定频率输出的正脉冲，可作为外部时钟或者外部定时脉冲使用。

3) PSEN: 外部程序内存读选通信号。在访问外 ROM 时，PSEN 会产生负脉冲信号 (即低电平信号) 作为外 ROM 的读选通信号。

4)  $\overline{EA}/VPP$ : 程序内存的控制信号。当  $\overline{EA}$  端保持低电平时, CPU 将只访问片外 ROM, 当  $\overline{EA}$  端保持高电平时, CPU 执行片内 ROM 指令 (除非程序计数器 PC 的值超过 0FFFH)。

(4) I/O 口引脚 (P0、P1、P2 和 P3)

STC89C51 单片机有 4 个 8 位并行输入/输出接口, 简称为 I/O 口。P0、P1、P2 和 P3 口共计 32 根输入/输出线。这 4 个接口可以并行输入/输出 8 位数据, 也可以按位输出, 即每一位可以独立输入/输出一个数据。



**知识链接**

在进行单片机应用系统设计时, 除了电源和地线引脚外, 以下引脚信号也必须连接相应电路。

1. 单片机最小系统电路。复位信号 RST 一定要连接复位电路, 外接晶体引线端 XTAL1 和 XTAL2 必须连接时钟电路, 这两部分是单片机能够工作所必须的电路。
2.  $\overline{EA}$  引脚一定要连接高电平或低电平。随着技术的发展, 单片机芯片内部的程序存储器空间越来越大, 因此, 用户程序一般都固化在单片机内部程序存储器中, 此时  $\overline{EA}$  引脚应接高电平。只有在使用内部没有程序存储器的 8031 芯片时,  $\overline{EA}$  引脚才接低电平, 该芯片目前已很少使用。

**3. 单片机最小系统**

所谓单片机最小系统是指单片机能进行正常工作的最简单电路, 包括单片机电路、电源电路、时钟电路和复位电路, 四者缺一不可。图 1-4 所示为单片机最小系统框图, 由于其结构简单、体积小、功耗低和成本低, 因此在简单的应用系统中得以广泛应用。

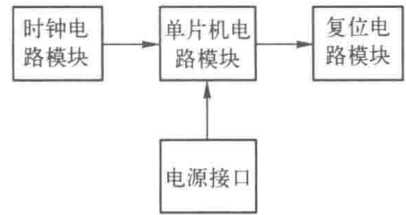


图 1-4 单片机最小系统框图

(1) 电源接口模块

单片机要能够正常运行工作, 就必须给以单片机上电, 其使用的电源为 +5V, 接入单片机的 VCC 引脚上, 同时在其单片机的 VSS 引脚上接地, 同时为了使单片机能有更好的抗干扰性, 所以一般在 VCC 和 VSS 之间接有高频和低频滤波的电容。

(2) 复位电路模块

复位是指使单片机内部各寄存器的值变为初始状态。单片机复位的条件: 当 RST (9 引脚) 端出现高电平并保持两个机器周期以上时, 单片机内部就会执行复位操作。复位电路有上电复位和按键复位两种实现方式, 如图 1-5 所示。

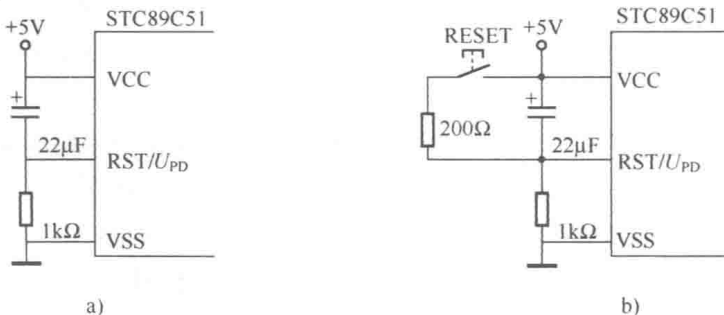


图 1-5 复位电路

1) 上电复位: 上电复位是指在单片机上电的瞬间, 单片机的 RST 端和 VCC 端电平相同, 随着电容的充电, 电容两端电压逐渐上升, RST 端电压逐渐下降, 完成复位。

2) 按键复位: 按键复位是指在单片机运行过程中, 按下复位键, RST 端电位变为高电平, 完成复位。

### (3) 时钟电路模块

要使单片机内部的每个部件之间协调一致的工作, 必须在时钟信号的控制下进行。单片机内部有一个用于构成振荡器的高增益放大器, 引脚 XTAL1 和 XTAL2 分别是此放大器的输入端和输出端, 只需要外接一个晶振和两个电容便构成自激振荡器, 为单片机系统提供时钟, 如图 1-6 所示。

时钟电路中的电容一般取值在 30pF 左右, 晶振的振荡频率范围一般为 1.2~24MHz, 而在通常情况下, 51 单片机使用的晶振频率为 6MHz 或 12MHz, 在通信系统中常用 11.0592MHz。

## 4. 单片机的时序周期

单片机系统的各部分是在 CPU 的统一指挥下协调工作的, CPU 微控制器根据不同指令, 产生相应的定时信号和控制信号, 各部分和各控制信号之间要满足一定时间顺序。

1) 振荡周期: 为单片机提供时钟信号的振荡源的周期 (晶振周期和外加振荡源周期)。

2) 状态周期: CPU 从一个状态转变到另一个状态所需的时间为状态周期。在 51 单片机中, 一个状态周期等于两个振荡周期。

3) 机器周期: 计算机完成一次完整的、基本的操作所需的时间为机器周期。51 单片机的一个机器周期为 6 个状态周期组成。

4) 指令周期: 执行完一条指令所需的时间。指令周期往往由一个或一个以上的机器周期所组成。51 单片机的指令周期一般有 1 个、两个和 4 个机器周期这 3 种周期状态。

以上 4 个周期之间的关系如图 1-7 所示。单片机的各周期的时间长短由单片机所外接的晶振振荡频率大小所决定, 例如本书中采用的是 12MHz 的晶振源, 那它的时钟周期是 1/12 (微秒), 相应它的一个机器周期是  $12 * (1/12)$ , 也就是 1 微秒。各周期的时间如下所示。

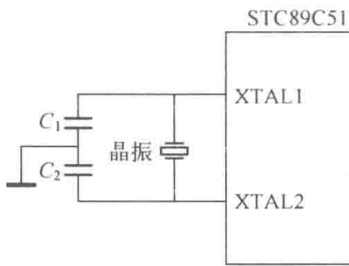


图 1-6 时钟电路

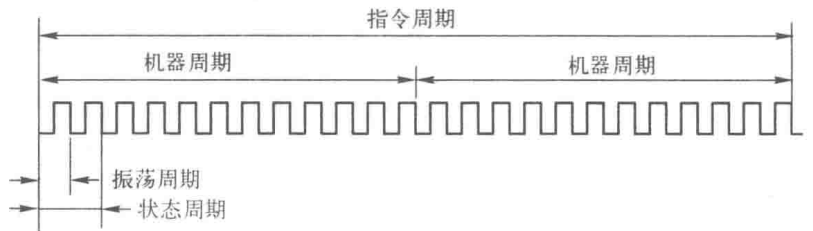


图 1-7 各周期的关系图

振荡周期=1/12 $\mu$ s, 状态周期=1/6 $\mu$ s, 机器周期=1 $\mu$ s, 指令周期=1~4 $\mu$ s

## 5. 单片机内部存储器

在单片机内部具有数据存储器 RAM 和程序存储器 ROM 两个内存存储资源。

### (1) 片内数据存储器

片内数据存储器又称为内部 RAM, 也称为随机存取内存, 主要用于数据缓冲和中间数

据的暂存，同时这种内存在使用过程中可随时进行写入和读取信息。STC89C51 单片机内部有 256 字节 (B)，通常把这 256 个字节分为两部分：低 128 字节 (单元地址 00H~7FH) 和高 128 字节 (单元地址 80H~0FFH)。

1) 片内低 128 字节 RAM。片内低 128 字节 RAM 按用途可分为 3 个区域，如图 1-8 所示。

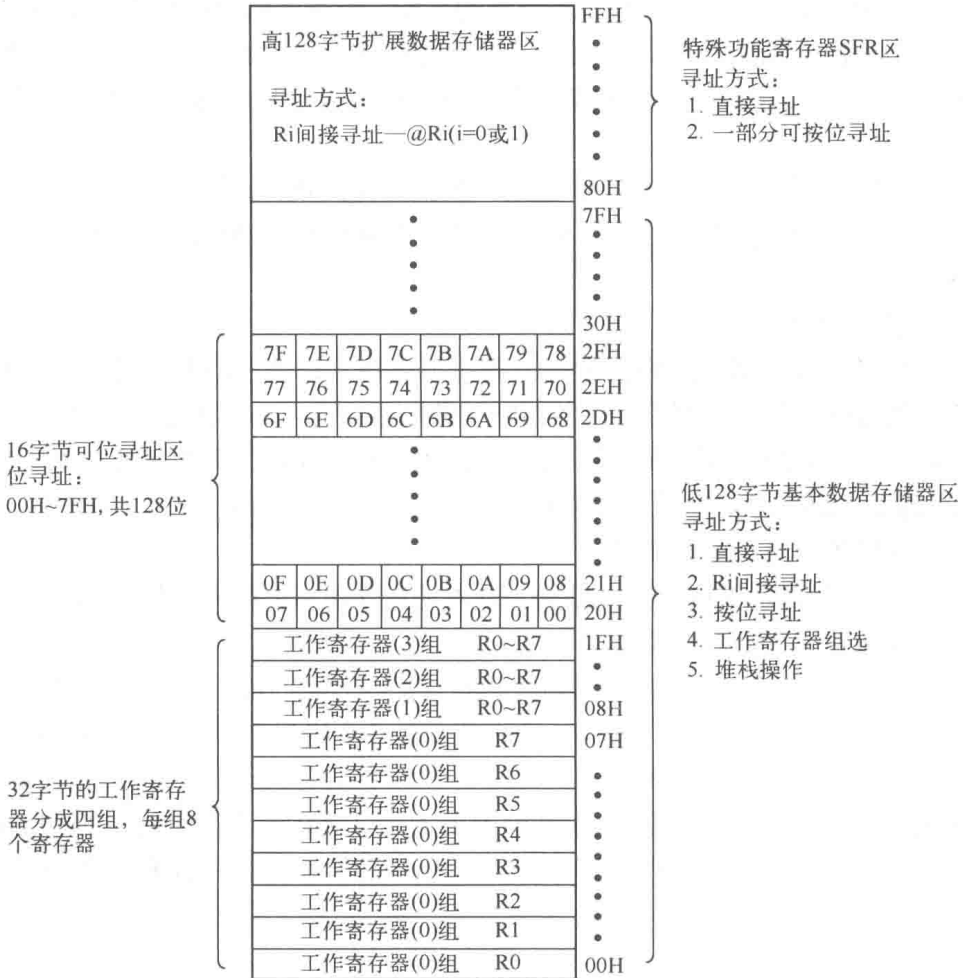


图 1-8 片内低 128 字节 RAM 结构图

① 通用寄存器区。地址为 00H~1FH 的空间单元为通用寄存器区，分为 4 组，每组 8 个单元，共 32 个单元。每组 8 个单元的符号为 R0~R7，如表 1-1 所示，CPU 通过程序状态字寄存器 (PSW) 中的 RS1 和 RS0 的状态选定选用哪组寄存器组工作。

表 1-1 工作寄存器组选择

RS1	RS0	工作寄存器组号	R0~R7 物理位元址
0	0	0	00H~07H
0	1	1	08H~0FH
1	0	2	10H~17H
1	1	3	18H~1FH

注：单片机复位后，RS0、RS1 的状态为 00。



### 知识链接

在单片机的 C 语言程序设计中，一般不会直接使用工作寄存器组 R0~R7。但是，在汇编语言设计中，工作寄存器组是用来直接存放操作数及中间结果等的重要寄存器。

② 位寻址区。地址为 20H~2FH 的空间单元为位寻址区，这 16 个单元（共 128 位）的每一位都有一个对应的地址，如表 1-2 所示。

表 1-2 位寻址区地址表

单元地址	MSB								位地址								LSB															
2FH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
2EH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
2DH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
2CH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
2BH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
2AH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
29H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
28H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
27H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
26H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
25H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
24H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
23H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
22H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
21H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H
20H	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H	77H	76H	75H	74H	73H	72H	71H	70H	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H	67H	66H	65H	64H	63H	62H	61H	60H

位寻址区的每一位都可当做软件触发器，由程序直接进行位处理。通常可以把各种程序状态标志、位控制变量存于位寻址区内。同时，位寻址区的单元也可进行按字节操作，作为一般的数据缓冲区使用。

③ 用户 RAM 区。地址为 30H~7FH 的 80 个单元空间用于供用户使用的 RAM 区。

2) 片内高 128 字节 RAM。内部数据存储器高 128 字节单元的地址为 80H~0FFH，在这 128 个单元中离散的分布着若干个特殊功能寄存器（简称为 SFR），专用于控制、选择、管理、存放单片机内部各部分的工作方式、条件、状态和结果。不同的 SFR 管理不同的硬件模块，负责不同的功能，包括程序状态字寄存器、累加器、I/O 口锁存器、定时/计数器、串口数据缓冲器及数据指针等，如表 1-3 所示。

表 1-3 特殊功能寄存器

寄存器名称	字节地址	MSB								位地址 (16 进制)								LSB								功能名称								
B	F0H	F7	F6	F5	F4	F3	F2	F1	F0	F7	F6	F5	F4	F3	F2	F1	F0	F7	F6	F5	F4	F3	F2	F1	F0	F7	F6	F5	F4	F3	F2	F1	F0	B 寄存器
ACC	E0H	E7	E6	E5	E4	E3	E2	E1	E0	E7	E6	E5	E4	E3	E2	E1	E0	E7	E6	E5	E4	E3	E2	E1	E0	E7	E6	E5	E4	E3	E2	E1	E0	累加器 A
PSW	D0H	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	程序状态字寄存器
		CY	AC	F0	RS1	RS0	OV	F1	P	CY	AC	F0	RS1	RS0	OV	F1	P	CY	AC	F0	RS1	RS0	OV	F1	P	CY	AC	F0	RS1	RS0	OV	F1	P	



(续)

寄存器名称	字节地址	位地址 (16 进制)								功能名称
		MSB							LSB	
IP	B8H	BF	BE	BD	BC	BB	BA	B9	B8	中断优先级控制寄存器
		/	/	/	PS	PT1	PX1	PT0	PX0	
P3	B0H	B7	B6	B5	B4	B3	B2	B1	B0	I/O 端口 3
		P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	
IE	A8H	AF	AE	AD	AC	AB	AA	A9	A8	中断允许控制寄存器
		EA	/	/	ES	ET1	EX1	ET0	EX0	
P2	A0H	A7	A6	A5	A4	A3	A2	A1	A0	I/O 端口 2
		P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	
SBUF	99H									串行数据缓冲器
SCON	98H	9F	9E	9D	9C	9B	9A	99	98	串行口控制寄存器
		SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
P1	90H	97	96	95	94	93	92	91	90	I/O 端口 1
		P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	
TH1	8DH									定时/计数器 1 (高字节)
TH0	8CH									定时/计数器 0 (高字节)
TL1	8BH									定时/计数器 1 (低字节)
TL0	8AH									定时/计数器 0 (低字节)
TMOD	89H	GAT	C/T	M1	M0	GAT	C/T	M1	M0	定时/计数器方式寄存器
TCON	88H	8F	8E	8D	8C	8B	8A	89	88	定时/计数器控制寄存器
		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
PCON	87H	SM0	/	/	/	/	/	/	/	电源控制寄存器
DPH	83H									数据指针 (高字节)
DPL	82H									数据指针 (低字节)
SP	81H									堆栈指针
P0	80H	87	86	85	84	83	82	81	80	I/O 端口 0
		P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	

注意：凡是特殊功能寄存器字节地址能被 8 整除的单元均能按位寻址。

① 程序计数器 PC。程序计数器 PC 是一个 16 位的计数器，其内容为将要执行的指令地址，寻址范围达 64KB。PC 有自动加 1 的功能，以实现程序的顺序执行。PC 没有地址，是不可寻址的，因此用户无法对它进行读写。但在执行转移、调用或返回等指令时能自动改变其内容，以改变程序的执行顺序。

② 累加器 ACC 及 B 寄存器。累加器 ACC 简称为 A，是所有特殊功能寄存器中最重要、使用频率最高的寄存器，常用于存放参加算术或逻辑运算的两个操作数中的一个，运算结果最终都存放在 A 中，许多功能也只有通过 A 才能实现。而 B 寄存器也是单片机内部特有的一个寄存器，主要用于乘法和除法运算，也可作为一般寄存器使用。

③ 数据指针 DPTR。数据指针 DPTR 是一个 16 位的专用寄存器，由 DPH (数据指针高

8 位) 和 DPL (数据指针低 8 位) 组成, 即可作为一个 16 位的寄存器使用, 也可作为两个独立的 8 位寄存器使用, 而 DPTR 通常用于存放外部数据存储器的存储单元地址。

④ 堆栈指针 SP。堆栈指针 SP 是一个 8 位的特殊功能寄存器, 用于指出堆栈栈顶的地址。数据被压堆栈时, SP 自动加 1; 数据出栈时, SP 自动减 1。

⑤ 程序状态字寄存器 PSW。程序状态字寄存器 PSW (8 位) 是一个标志寄存器, 用于存放指令运行结果的状态信息, 以供程序查询和判别。有些位在执行中硬件自动设置, 而有些位由用户自行设定。STC89C51 单片机中的 PSW 寄存器各位的含义如表 1-4 所示。

表 1-4 PSW 寄存器各位的含义

PSW(0D0H)	D7	D6	D5	D4	D3	D2	D1	D0
位定义	Cy	Ac	F0	RS1	RS0	OV	—	P
位含义	进位标志	辅助进位标志	用户自定义位	工作寄存器选择位		溢出标志	未定义位	奇偶标志
位地址	0D7H	0D6H	0D5H	0D4H	0D3H	0D2H	0D1H	0D0H

**Cy:** 进位标志。在执行加法、减法指令时, 若运算结果的最高位 (D7 位) 有借位时, Cy 位被置“1”, 否则被清“0”。

**Ac:** 辅助进位标志位。主要用于存放 BCD 码加法调整。在执行加、减法指令时, 其低四位向高四位有进位或借位时 (D3 位向 D4 位), AC 位被置“1”, 否则清“0”。

**F0:** 用户定义标志位。可通过位操作指令将其置位、清零。

**RS1、RS0:** 工作寄存器组选择位。用于选择单片机内部 4 组工作寄存器中的哪一组工作。RS0、RS1 状态与工作寄存器 R0~R7 的物理地址关系如表 1-1 所示。

**OV:** 溢出标志位。在计算机内, 带符号的数一律用补码表示。在 8 位二进制中, 补码所能表示的范围是-128~+127, 当运算结果超出这个范围时, OV 标志被置“1”, 否则清“0”。

**P:** 奇偶标志位。用于指示累加器 ACC 在运算结果中 1 的个数的奇偶性, 当累加器 A 中“1”的个数为奇数, 则 P 为被置“1”; 否则, 当累加器 A 中“1”的个数为偶数时, P 位被清“0”。

⑥ I/O 口寄存器。单片机内部有 4 个 I/O 口寄存器 P0、P1、P2 和 P3, 实际上就是 P0~P3 口所对应的 I/O 口锁存器, 用于锁存通过端口的数据。

当单片机复位后, 各特殊功能寄存器的状态如表 1-5 所示。

表 1-5 寄存器复位状态

寄存器	复位值	寄存器	复位值
PC	0000H	TMOD	00H
B	00H	TCON	00H
PSW	00H	TH0	00H
SP	07H	TL0	00H
ACC	00H	TH1	00H
DPTR	0000H	TL1	00H
P0~P3	0FFH	SCON	00H
IE	0XX00000B	SBUF	不定
IP	XXX00000B	PCON	0XXX0000B

## (2) 片内程序存储器

片内程序存储器主要是用来存放计算机中所事先编制好的程序和表格常数。在 STC89C51 单片机芯片中有 4KB 的片内程序内存单元，其地址为 0000H~0FFFH，其中地址为 0003H~002AH 的单元在使用时是有特殊规定的。

地址为 0000H~0002H 的 3 个单元是系统的启动单元，在单片机进入复位后，单片机会自动从 0000H 单元开始执行指令程序。但实际上，这 3 个单元并不能存放任何完整的程序，用户在使用时，必须在该单元存放一条无条件跳转指令，以便跳转到指定的程序地址处执行。

地址为 0003H~002AH 的 40 个单元则被平均分为 5 段，每段 8 个单元，分别用做 5 个中断源的中断地址区，具体分布如表 1-6 所示。

表 1-6 中断源地址分配表

中断入口地址	中断地址区间	中 断 源
0003H	0003H~000AH	外部中断 0
000BH	000BH~0012H	定时/计数器中断 T0
0013H	0013H~001AH	外部中断 1
001BH	001BH~0022H	定时/计数器中断 T1
0023H	0023H~002AH	串行口中断

### 1.1.3 认知单片机编程语言

单片机的编程语言大致可分为机器语言、汇编语言和高级语言 3 种。本书以汇编语言和高级语言中常用的 C 语言作为单片机的编程语言进行讲解。

1) 机器语言是由机器能直接识别的由 0 和 1 组成的编码（通常用十六进制数表示），也被称为机器指令。用机器指令编写的程序称为机器语言源程序，它是机器所能理解和执行的，但是人们记忆和读写都很困难。

2) 汇编语言是一种用“助记符”和数字符号来表示机器指令的符号语言，是最接近于机器码的一种语言。但是它必须通过汇编程序汇编成机器语言程序后，机器才能理解和执行，汇编过程也可通过手工完成。其主要优点是占用资源少，程序执行效率高。由于它一条指令就对应一条机器码，每一步的执行动作都很清楚，并且程序大小和堆栈调用情况都容易控制，调试起来也比较方便。

为了使程序的结构清晰明了，方便程序的修改及补充等，一般 51 单片机汇编语言程序可以以按照如下框架书写，其中各项详细说明将在后续的各个项目里陆续展开介绍。

```
ORG    0000H    ; 程序初始化入口地址
LJMP   MAIN    ; 程序跳入主函数 MAIN 中
ORG    0003H    ; 外部中断 0 程序入口地址
LJMP   INT_0   ; 程序跳入中断子程序 INT_0 中
ORG    000BH    ; 定时/计数器 T0 程序入口地址
LJMP   T_0     ; 程序跳入中断子程序 T_0 中
ORG    0013H    ; 外部中断 1 程序入口地址
LJMP   INT_1   ; 程序跳入中断子程序 INT_1 中
ORG    001BH    ; 定时/计数器 T1 程序入口地址
LJMP   T_1     ; 程序跳入中断子程序 T_1 中
ORG    0023H    ; 串口中断程序入口地址
```